研究生教学用书
公共基础课系列

数值分析

（第2版）

Numerical Analysis

李 红

华中科技大学出版社
http://www.hustp.com

研究生教学用书
公共基础课系列

数值分析

（第 2 版）

李　红

华中科技大学出版社
中国·武汉

图书在版编目(CIP)数据

数值分析(第2版)/李 红 .—武汉:华中科技大学出版社,2010年5月
ISBN 978-7-5609-3062-6

Ⅰ.数… Ⅱ.李… Ⅲ.数值计算 Ⅳ.O241

中国版本图书馆CIP数据核字(2010)第084617号

数值分析(第2版) 李 红

策划编辑:李 德
责任编辑:李 德 封面设计:潘 群
责任校对:张 琳 责任监印:熊庆玉

出版发行:华中科技大学出版社(中国·武汉)
武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉众欣图文激光照排中心
印 刷:华中科技大学印刷厂

开本:787mm×960mm 1/16 印张:18.5 字数:350 000
版次:2010年5月第2版 印次:2010年5月第3次印刷 定价:29.80元
ISBN 978-7-5609-3062-6/O·296

第 2 版前言

本书第 2 版是应新世纪对理工科学生的科学计算能力的要求，在《数值分析》（第 1 版）（华中科技大学出版社出版）的基础上，广泛吸取校内外教师的意见后修订而成的. 从这几年使用第 1 版教材的高校教师的反馈情况来看，普遍认为本教材取材合理、叙述清楚、简明精要、易于教学. 因此，新版在主要内容和结构框架上未作大的改动，但从教学角度出发对语句进行了仔细的推敲，改写了一些陈述，调整了例、习题的配置.

对曾使用过第 1 版教材的各位教师和读者表示衷心的感谢，正是依据你们使用后的意见，作者在第 2 版中修正了不少错漏，使得本书更趋完善.

新版中的附录包含了各章实验题的 Matlab 程序源代码，所有的程序均在 Matlab R 2008a 中通过测试. 肖光润博士完成了教材中整个数值实验的程序设计工作. 在此，谨致以诚挚的谢意.

限于水平，书中一定有不少缺点和错漏之处，恳望得到批评指正.

作　者

2010 年 4 月于华中科技大学

前　言

21世纪是数字化、信息化、智能化技术超常规发展的年代，计算机作为其基本的载体，其技术理论的研究和应用直接影响到当今科学技术发展的速度和质量.

随着计算机的迅速发展，熟练地运用计算机进行科学计算已成为科技工作者不可缺少的技能.应用计算机求解各类问题更成为各行各业知识更新的必要环节，培养学生和应用者的科学计算能力，数值分析课程具有不可替代的作用.

作者经过十多年的教学实践积累和不断的探索改进，重新编著了《数值分析》教材.本书以提高学生的数学素质，培养学生科学计算的实际能力为要旨，强调重概念、重方法、重应用及重能力的培养，其特点是:(1) 内容新，删除了以前教材中的陈旧内容，增加科学计算中一些新的使用方法和计算方法;(2) 在着重数值计算基本原理和各种方法的基本思想阐述时，注重强化数学概念的严密性和准确性，使学生的数学逻辑思维能力得到训练;(3) 加强数值实验，强化实践能力的培养.本书各章都给出相应的数值实验习题，以帮助学生掌握各种数值计算方法，从而提高学生应用数值计算方法解决实际问题的能力.

在编写本教材的过程中，得到了我校研究生院及数学系的大力支持，很多老师给予了作者关心和指教.徐长发教授、贺云峰老师在试用教材期间提出了宝贵意见，徐永兵老师、郑成勇硕士完成了教材中整个数值实验的程序设计工作.在此，谨致以诚挚的谢意.

限于水平，书中一定有不少缺点和错漏之处，恳望得到批评指正.

作　者

2003.8

目　录

第1章　绪　　论

1.1　课程的意义、内容和特点

计算机技术是对当今科学、工程技术和人类社会生活影响最深刻的最新技术之一，它对科学技术最深刻的影响，莫过于使科学计算平行于理论分析和实验研究，成为人类探索未知科学领域和进行大型工程设计的方法和手段. 在独创性工作的先行性研究过程中，科学计算更具有突出的作用. 科学计算能力是跨世纪人才不可或缺的. 科学计算是数学与计算机有机结合的结果，其核心内容是以现代化的计算机及数学软件为工具，以数学模型为基础进行模拟研究. 近年来，它也成为数学科学本身发展的源泉和途径之一，高等教育中如何培养学生科学计算的能力的问题正日益受到关注，成为当前教育改革的核心和焦点之一.

数值分析及有关的数学基础教学环节，在培养学生科学计算能力上具有不可替代的作用. 目前，国内许多高等学校已将数值分析列入自然科学、工程技术乃至社会科学本科的教学计划之中.

本课程的内容可概括为用计算机求解数学问题的数值方法和理论，简称数值分析或计算方法. 数学学科十分广泛，这里只涉及工程和科学实验中常见的数学问题，如线性方程组、函数的插值、离散数据的拟合、微积分、微分方程、非线性方程等，它们是其他数学学科的基础.

电子计算机实质上只会做有限次的加、减、乘、除等基本运算，而研究怎样通过计算机所能执行的基本运算，来求得各类问题的数值解或近似数值解，就是数值分析所要研究的根本课题. 由基本运算及运算顺序的规定所构成的完整的解题步骤，称为算法. 数值分析的根本任务，也可说是研究算法.

有许多数学问题的解，不可能经过有限次算术运算计算出来. 例如，计算任意角的三角函数值、求一般方程的根、计算任意函数的积分、求一般微分方程的解. 对于这类问题，计算方法常常采用近似替代的办法.

例如，已知角的弧度 x 在 0 与 $\pi/4$ 之间，计算 $\sin x$ 的值.

根据微分学的 Taylor(泰勒)公式，可得

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + R_{2n+1}(x).$$

由于计算多项式的值只用到算术运算，而且当 n 充分大时，余项 $R_{2n+1}(x)$ 的数值很

小,便可用上式右端前$(2n+1)$次多项式来近似替代 $\sin x$,把这多项式的值当作 $\sin x$ 的值,得到计算 $\sin x$ 的近似公式

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}.$$

这个公式在 $x=0$ 的邻近近似程度较高,但在其他地方,这个公式误差较大.近似值与真正值之差,称为该近似值的误差.理想的公式,应使误差在整个区间$[0,\pi/4]$上都很小.计算机常用的标准函数,就是按这种“理想”公式设计的.

又例如,求非线性方程 $f(x)=0$ 的根.一般说方程很难求解,但如已知根的粗略近似值为 x_0,则根据 Taylor 级数

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots$$

取等式右边前两项近似替代 $f(x)$,就会得到很容易求解的线性方程

$$f(x_0) + f'(x_0)(x-x_0) = 0.$$

把解出的 x 记为 x_1,则有

$$x_1 = x_0 - f(x_0)/f'(x_0).$$

x_1 虽然不一定是根,但往往比 x_0 更接近于根.用 x_1 代替上面的 x_0,进行类似计算,即可得 x_2.这样继续下去,往往可得一系列越来越接近根的近似值 $x_1,x_2,\cdots$,其中

$$x_{n+1} = x_n - f(x_n)/f'(x_n), \tag{1.1}$$

这种求根法称为牛顿迭代法.以具体方程 $x^2-2=0$ 为例,式(1.1)成为

$$x_{n+1} = x_n - \frac{x_n^2-2}{2x_n} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right).$$

取 $x_0=1.4$,按上式计算得 $x_1=1.414285714$,$x_2=1.414213564$,可见越来越接近根$\sqrt{2}$ $(=1.41421356237\cdots)$.

再如求解常微分方程的初值问题

$$\begin{cases} y' = f(x,y), & a \leqslant x \leqslant b, \\ y(a) = y_0, \end{cases} \tag{1.2}$$

通常无法求出区间$[a,b]$上的解 $y(x)$的解析表达式.但实际问题往往只需算出它在某些点处的近似值,如在点 $x_n=a+nh$ $(n=0,1,2,\cdots,N;h=(b-a)/N)$处的近似值 $y_n \approx y(x_n)$.由式(1.2),得

$$y'(x_n) = f(x_n, y(x_n)), \quad n=0,1,2,\cdots,N. \tag{1.3}$$

又由于

$$y'(x_n) = \lim_{h\to 0} \frac{y(x_{n+1}) - y(x_n)}{h},$$

用差商$[y(x_{n+1})-y(x_n)]/h$ 近似替代式(1.3)中的微商(导数)$y'(x_n)$,用 y_n 近似替代 $y(x_n)$,得到

$$\frac{y_{n+1}-y_n}{h} = f(x_n, y_n), \quad n = 0,1,2,\cdots,N-1,$$

从而有

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0,1,2,\cdots,N-1. \tag{1.4}$$

这样，由已知值 y_0 出发，便可逐步算出 $y_1, y_2, \cdots, y_N$，这种求初值问题(1.2)解近似值的方法，称为 Euler(欧拉)折线法. 以具体问题

$$\begin{cases} y' = y - \dfrac{2x}{y}, & 0 \leqslant x \leqslant 1, \\ y(0) = 1 \end{cases}$$

为例，则 Euler 法计算公式(1.4)成为 $y_{n+1} = y_n + h(y_n - 2x_n/y_n)$. 取 $h=0.1$ 及 $h=0.01$，由 $y_0 = 1$ 出发，按此公式可算得表 1.1 中的第二、三行；第四行是解的真正值 $y(x_n) = \sqrt{1+2x_n}$. 计算时用 5 位小数，但表中只记录了 4 位小数. 由表1.1可见，h 越小，计算结果越准确.

表 1.1

x_n	0.1	0.2	0.3	0.4	0.5	1
y_n	1.1	1.1918	1.2774	1.3582	1.4351	1.7848
y_n	1.0959	1.1841	1.2662	1.3434	1.4164	1.7380
$y(x_n)$	1.0954	1.1832	1.2649	1.3416	1.4142	1.7321

上述三例都是利用近似替代，把不能用有限次运算求解的问题，转化为比较简单的，可用有限次运算求解的问题，从而用简化问题的解作为原来问题的近似解. 由此，人们自然会问，这种近似解的误差有多大？能否通过增大 n(如前两例)或缩小 h(如第三例)使误差无限减小？数值分析在采用近似替代的同时，还需研究这两个问题，即误差估计和收敛性问题.

随着电子计算机的发展，计算机运算速度越来越快，存储数据的容量越来越大. 但无论多快多大，毕竟有一定限度. 所以，对于可用有限次运算求解的问题，还需考虑先算什么量，后算什么量，才能最快算出结果，并使占有的内存最省.

例如，计算多项式

$$0.0625x^4 + 0.425x^3 + 1.215x^2 + 1.912x + 2.1296$$

的值. 如果先算各项然后相加，需做 10 次乘法和 4 次加法. 但如改为下式

$$(((0.0625x + 0.425)x + 1.215)x + 1.912)x + 2.1296$$

计算，则只需做 4 次乘法和 4 次加法(此算法称为秦九韶算法). 进而改为下式

$$[(0.5x + 0.6)^2 + 0.5x + 0.7][(0.5x + 0.6)^2 + 0.8] + 0.9$$

计算，则可减少到 3 次乘法和 5 次加法.

又例如，要求解 $n=20$ 的线性代数方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \qquad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n, \end{cases}$$

如果按 Cramer(克莱姆)法则求解,即按下面公式计算:

$$x_k = D_k/D, \quad k = 1,2,\cdots,n,$$

其中,$D=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$,$D_k$ 是把 D 中第 k 列$(a_{1k},a_{2k},\cdots,a_{nk})^{\mathrm{T}}$ 换为$(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$ 后得到的行列式.

那么,这要计算$(n+1)$个行列式,并做 n 次除法,而每个行列式包含 $n!$ 个乘积,计算每个乘积需做$(n-1)$次乘法,这样共需要$[(n+1)!\,(n-1)+n]$次乘除法.当 $n=20$ 时,则要做 9.7×10^{20} 次乘除法;即使在每秒做 10^8 次乘除法的计算机上计算,也需 30 多万年! 可见理论上很"漂亮"的 Cramer 法则在计算机上并不适用.

因此,在构造算法时,还应考虑如何计算,才能既快又省.

1.2 误差及有关概念

前述已提及过误差,这里对误差及有关概念作进一步的讨论.

1.2.1 误差的来源

利用计算机解决科学计算问题的过程大致可以分为两个环节.首先,将实际问题归结为数学问题,建立起比较适合的、具体的数学模型,如微分、积分、方程、级数求和等;然后,选择适当的解题方法,编制好程序,上机算出结果.在这两个环节中,每一步都可能产生误差.

一个物理量的真实值和计算出的值往往存在差异,它们之差称为误差.

误差大致可分为以下四类.

(1)模型误差.通过对实际问题进行抽象,简化得到的数学模型与实际现象之间必然存在误差,这种误差称之为模型误差.

(2)观测误差(参量误差、数据误差).一般数学问题包含若干参量,它们的值往往通过观测得到,而观测难免带来误差,这种误差称之为观测误差.

(3)截断误差(方法误差).一般数学问题难以求解,往往要通过近似替代,简化为较易求解的问题,由简化引起的误差称之为截断误差.

许多数学运算(如微分、积分及无穷级数求和等)是通过极限过程来定义的,然而计

算机上只能完成有限次的算术运算和逻辑运算，因此需要将解题方案加工成算术运算与逻辑运算的有限序列，这样的加工常常表现为无穷过程的截断，由此产生的误差通常称作截断误差. 如 e^x 可展开为级数形式

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots.$$

但用计算机求值时，不能直接得出等号右端无穷多项的和，而只能截取有限项求值，即

$$S_n(x) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!},$$

用部分和 $S_n(x)$ 作为 e^x 的值必然会有误差，由 Taylor 余项定理，其截断误差为

$$e^x - S_n(x) = \frac{x^{n+1}}{(n+1)!}e^{\theta x}, \quad 0 < \theta < 1.$$

(4)舍入误差(计算误差). 由于实际计算是按有限位数进行的，数值解的每一步都可能有误差，这种误差称之为舍入误差.

在计算过程中所用数据可能位数很多，然而受机器字长的限制，用机器代码表示的数据必须舍入成一定的位数，这也会引起舍入误差.

例如，在实际计算时用 0.33333 代替了 $\frac{1}{3}$，则舍入误差为

$$R = \frac{1}{3} - 0.33333 = 0.00000333\cdots.$$

数值分析研究数学问题的数值解法，所以不讨论模型误差，只讨论截断误差与舍入误差.

1.2.2 误差限与有效数字

准确数 x 与其近似数 x^* 之差，即 $e = x - x^*$ 称为绝对误差.

误差虽然不可避免，但人们总是希望计算结果能足够准确，这就需要估计误差，误差 $(x - x^*)$ 的具体数值通常是无法确定的，只能根据测量工具或计算过程，设法估算出它的取值范围，即误差绝对值的一个上界

$$|e| = |x - x^*| \leqslant \varepsilon.$$

这种上界 ε 称作近似值 x^* 的绝对误差限，简称误差限，或称精度(误差限一般都取到某位的半个单位). 精度为 ε 的近似值 x^*，可以认为和真值 x 关于允许误差 ε 是"重合"的，或者说，结果关于精度 ε 是"准确"的.

写出一个近似值后，当然希望指明它的准确程度，为此需要引进有效数字的概念.

众所周知，当准确数 x 有很多位数时，常常按"四舍五入"原则去得到 x 的近似数，例如，$\pi = 3.14159265\cdots$ 按舍入原则若取四位小数，得 $\pi = 3.1416$；取五位小数，得 $\pi = 3.14159$，它们的绝对误差不超过末位数的半个单位，即

$$|\pi - 3.1416| \leqslant \frac{1}{2} \times 10^{-4}, \quad |\pi - 3.14159| \leqslant \frac{1}{2} \times 10^{-5}.$$

四以下舍,五以上入这些都无问题,若刚好是五,则作如下规定:若前面是偶数,则将五舍去;若前面是奇数,则将五进一. 实践证明,当进行大量运算时,按上述原则进行舍入,整个运算的误差积累较小.

如果近似值 x^* 的误差限是它的某一位的半个单位,就说它"准确"到这一位,并且从这一位一直到前面第一个非零数字为止的所有数字均称有效数字,有效数字实际上是将四舍五入抽象成数学语言而引入的一个新名词.

定义 1.1 若近似值 x^* 的误差限是某一位的半个单位,该位到 x^* 的第一位非零数字有 n 位,则称 x^* 有 n 位有效数字.

定义 1.1 也可如下表述.

定义 1.1′ 对于 x 的近似值(规格化形式)

$x^* = \pm 0.a_1a_2\cdots a_n \times 10^m$(其中,$a_1,a_2,\cdots,a_n$ 是 0 到 9 之间的自然数,$a_1 \neq 0$),

如果误差
$$|x-x^*| \leqslant \frac{1}{2}\times 10^{m-l},\quad 1\leqslant l\leqslant n,$$
则称近似值 x^* 有 l 位有效数字.

按照这种说法,π 的近似值 3.1416,3.14159 分别有 5 位和 6 位有效数字.

1.2.3 相对误差限与有效数字的联系

用绝对误差来刻画近似数的精确程度是有局限性的,因为它没有反映出它在原数中所占的比例. 于是,记
$$e_r = \frac{x-x^*}{x},$$
e_r 称为近似值 x^* 的相对误差,由于准确值 x 未知,实际上总把 $\frac{x-x^*}{x^*}$ 作为 x^* 的相对误差,相对误差一般用百分比表示.

相对误差绝对值的一个上限 ε_r^* 称作近似值的相对误差限,即有
$$\frac{|x-x^*|}{|x^*|} \leqslant \varepsilon_r^*.$$
下面分析相对误差限与有效数字的联系.

定理 1.1 设近似值 $x^* = \pm 0.a_1a_2\cdots a_n\times 10^m$ 有 n 位有效数字,则其相对误差限为 $\frac{1}{2a_1}\times 10^{-n+1}$.

证 由于 x^* 有 n 位有效数字,则
$$|x-x^*| \leqslant \frac{1}{2}\times 10^{m-n},$$
而
$$|x^*| \geqslant a_1\times 10^{m-1},$$
所以
$$\frac{|x-x^*|}{|x^*|} \leqslant \frac{\frac{1}{2}\times 10^{m-n}}{a_1\times 10^{m-1}} = \frac{1}{2a_1}\times 10^{-n+1}. \tag{1.5}$$

定理 1.2 设近似值 $x^*=\pm 0.a_1a_2\cdots a_n\times 10^m$ 的相对误差限为 $\dfrac{1}{2(a_1+1)}\times 10^{-n+1}$，则它至少有 n 位有效数字.

证 由于

$$|x^*|\leqslant (a_1+1)\times 10^{m-1},$$

故由题设，有

$$\begin{aligned}|x-x^*|&=\frac{|x-x^*|}{|x^*|}\cdot|x^*|\leqslant\frac{1}{2(a_1+1)}\times 10^{-n+1}\times(a_1+1)\times 10^{m-1}\\&=\frac{1}{2}\times 10^{m-n}.\end{aligned}\tag{1.6}$$

所以 x^* 至少有 n 位有效数字.

1.2.4 数值运算的误差估计

设近似数 x_1^* 与 x_2^* 的误差限分别为 $\varepsilon(x_1^*)$ 及 $\varepsilon(x_2^*)$，则它们进行加、减、乘、除运算得到的误差限分别为

$$\begin{cases}\varepsilon(x_1^*\pm x_2^*)\approx\varepsilon(x_1^*)+\varepsilon(x_2^*),\\ \varepsilon(x_1^*\cdot x_2^*)\approx|x_1^*|\,\varepsilon(x_2^*)+|x_2^*|\,\varepsilon(x_1^*),\quad (x_2^*\neq 0).\\ \varepsilon(x_1^*/x_2^*)\approx\dfrac{|x_1^*|\,\varepsilon(x_2^*)+|x_2^*|\,\varepsilon(x_1^*)}{|x_2^*|^2}\end{cases}\tag{1.7}$$

设函数为 $f(x_1,x_2,\cdots,x_n)$，若需要计算 $A=f(x_1,x_2,\cdots,x_n)$，而 $x_1,x_2,\cdots,x_n$ 的近似值为 $x_1^*,x_2^*,\cdots,x_n^*$，A 的近似值为 $A^*=f(x_1^*,x_2^*,\cdots,x_n^*)$，于是由 Taylor 展开得近似值 A 的误差 $e(A^*)$ 为

$$\begin{aligned}e(A^*)=A-A^*&=f(x_1,x_2,\cdots,x_n)-f(x_1^*,x_2^*,\cdots,x_n^*)\\&\approx\sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,x_2^*,\cdots,x_n^*)}{\partial x_k}\right)(x_k^*-x_k)\\&=\sum_{k=1}^{n}\left(\frac{\partial f}{\partial x_k}\right)^*e_k^*,\end{aligned}\tag{1.8}$$

误差限为

$$\varepsilon(A^*)\approx\sum_{k=1}^{n}\left|\left(\frac{\partial f}{\partial x_k}\right)^*\right|\varepsilon(x_k^*),\tag{1.9}$$

而 A^* 的相对误差限为

$$\varepsilon_r^*=\varepsilon_r(A^*)=\frac{\varepsilon(A^*)}{|A^*|}\approx\sum_{k=1}^{n}\left|\left(\frac{\partial f}{\partial x_k}\right)^*\right|\frac{\varepsilon(x_k^*)}{|A^*|}.\tag{1.10}$$

1.3 数值稳定性和病态问题

算法的数值稳定性是指，用一个算法进行计算，由于初始数据误差在计算中传播，使计算结果误差增长很快，就说该算法是数值不稳定的，否则是数值稳定的.

例 1.1　求积分 $I_n=\int_0^1 \frac{x^n}{x+5}\mathrm{d}x,\quad n=0,1,\cdots,8$ 的值.

解　由

$$I_n+5I_{n-1}=\int_0^1 \frac{x^n+5x^{n-1}}{x+5}\mathrm{d}x=\int_0^1 x^{n-1}\mathrm{d}x=\frac{1}{n},$$

可得两个递推算法

算法一　$I_n=\frac{1}{n}-5I_{n-1},\quad n=1,2,\cdots,8,$

算法二　$I_{n-1}=\frac{1}{5}\left(\frac{1}{n}-I_n\right),\quad n=8,7,\cdots,1.$

算法一的初始值

$$I_0=\int_0^1 \frac{1}{x+5}\mathrm{d}x=\ln(1.2)=0.1823215\cdots.$$

算法二的初始值用 Matlab 的库函数 quadl 计算的结果是

$$I_8=\int_0^1 \frac{x^8}{x+5}\mathrm{d}x=0.01883692\cdots.$$

现在假设计算机的十进制尾数是 4 位，即取十进制 4 位有效数字进行计算. 记算法一的计算值为 $\tilde{I}_n^{(1)}$，算法二的计算值为 $\tilde{I}_n^{(2)}$. 运算结果见表 1.2，其中 $\tilde{I}_n$ 是由库函数 quadl 求出的 I_n 的近似值，取 6 位有效数字. 易知，对于任何自然数 n，都有 $0<I_n<1$，并且 I_n 单调减. 可见，算法一是不稳定的，算法二是稳定的.

表 1.2

n	$\tilde{I}_n^{(1)}$	$\tilde{I}_n^{(2)}$	$\tilde{I}_n$
0	0.1823	0.1823	0.182322
1	0.08850	0.08839	0.0883922
2	0.05750	0.05804	0.0580389
3	0.04583	0.04314	0.0431387
4	0.02085	0.03431	0.0343063
5	0.09575	0.02847	0.0284684
6	−0.3121	0.02433	0.0243249
7	1.703	0.02123	0.0212326
8	−8.390	0.01884	0.0188369

由于串行运算的舍入误差是逐步传递的，所以可用“向后误差”分析方法，即把舍入误差的影响看成是初值误差即输入数据误差的传播.

定义 1.2　对于某个算法，若输入数据的误差在计算过程中迅速增长而得不到控制，则称该算法是数值不稳定的，否则是数值稳定的.

在例 1.1 中，当仅考虑初始值有误差时，对于算法一，由

$$I_n=\frac{1}{n}-5I_{n-1},\quad \tilde{I}_n^{(1)}=\frac{1}{n}-5\tilde{I}_{n-1}^{(1)},$$

得知误差 $e_n^{(1)}=I_n-\tilde{I}_n^{(1)}$ 满足

$$e_n^{(1)}=-5e_{n-1}^{(1)}=(-5)^n e_0^{(1)},\quad n=1,2,\cdots,$$

因此算法一是不稳定的. 对于算法二,同理可知误差 $e_i^{(2)}=I_i-\tilde{I}_i^{(2)}$ 满足

$$e_{i-1}^{(2)}=-\frac{1}{5}e_i^{(2)},\quad i=n,n-1,\cdots,1,$$

从而 $e_0^{(2)}=\left(-\frac{1}{5}\right)^n e_n^{(2)}$,因此算法二是稳定的.

例 1.2 求实系数二次代数方程 $ax^2+bx+c=0$ 的根,其中 $b^2-4ac>0,a\neq0$.

解 考虑下列两种算法.

算法一
$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}.$$

算法二
$$x_1=\frac{-b-\mathrm{sign}(b)\sqrt{b^2-4ac}}{2a},\quad x_2=\frac{c}{ax_1},$$

其中 sign 表示取数的符号,即

$$\mathrm{sign}(b)=\begin{cases}1, & b\geqslant 0,\\ -1, & b<0.\end{cases}$$

对于算法一,若 $b^2\gg 4ac$,则算法一是不稳定的,否则是稳定的. 这是因为前一种情况的分子有一个相近数相减,会大量损失有效数字,从而必有一个结果的误差很大. 算法二不存在这个问题,因此在任何情况下都是稳定的. 称算法一是条件稳定的,算法二是无条件稳定的.

与例 1.1 一样用 4 位有效数字计算方程

$$x^2+62.10x+1.000=0,$$

结果是

算法一 $x_1\approx-62.08,\quad x_2\approx-0.02000.$

算法二 $x_1\approx-62.08,\quad x_2\approx-0.01611.$

准确解是 $x_1=-62.083892\cdots$, $x_2=-0.016107237\cdots$,这里 $b^2\gg 4ac$,所以算法一不稳定,舍入误差对 x_2 的影响很大.

1.4 数值运算中的一些原则

数值运算总是在一个预先设计好的算法中进行的. 所谓算法,就是一个有限的基本运算序列,这个序列规定了怎样从输入数据去计算出问题的解. 由于运算是在计算机上进行的,而计算机的字长有限,因而产生舍入误差. 为减小舍入误差的影响,设计算法时应遵循以下一些原则.

(1) 避免除数绝对值远远小于被除数绝对值的除法(见式(1.7)中的除法运算).